

## Synonym

As a language, C also has synonyms. For example, in C array and pointer are interchangeable. Programme 1 shows four different ways to access an element of a matrix, through different words which mean the same thing. Notice that indices run beginning from zero not one.

### **Programme 1** *Array and pointer representations of a matrix*

```

1 /* synonyms, Kit Tyabandha, 6 Feb 07 */
2 #include <stdio.h>
3 int
4 main(){
5     int a[5][6]={
6         {11, 12, 13, 14, 15, 16},
7         {21, 22, 23, 24, 25, 26},
8         {31, 32, 33, 34, 35, 36},
9         {41, 42, 43, 44, 45, 46},
10        {51, 52, 53, 54, 55, 56}}, i=2, j=3;
11    printf("\n");
12    printf("a[i][j] =\n\t\
13        %d, %d, %d, %d, %d, %d\n\t\
14        %d, %d, %d, %d, %d, %d\n\t\
15        %d, %d, %d, %d, %d, %d\n\t\
16        %d, %d, %d, %d, %d, %d\n\t\
17        %d, %d, %d, %d, %d, %d\n\n",\
18        a[0][0], a[0][1], a[0][2], a[0][3], a[0][4], a[0][5],\
19        a[1][0], a[1][1], a[1][2], a[1][3], a[1][4], a[1][5],\
20        a[2][0], a[2][1], a[2][2], a[2][3], a[2][4], a[2][5],\
21        a[3][0], a[3][1], a[3][2], a[3][3], a[3][4], a[3][5],\
22        a[4][0], a[4][1], a[4][2], a[4][3], a[4][4], a[4][5]);
23    printf("a[%d][%d] = %d\n\n", i, j, a[i][j]);
24    printf("*(a[%d]+%d) = %d\n\n", i, j, *(a[i]+j));
25    printf("(*(a+%d)+%d) = %d\n\n", i, j, (*(a+i)+j));
26    printf("(*(a+%d*n+%d) = %d\n\n", i, j, (*(a+i*6+j));
27    return 0;
28 }
```

We compile and run this programme as the following.

```

kit@nebula:~/prog/c$ make tst
gcc -c -g -Wall tst.c
gcc -g tst.o -o tst -lcurses -ldl -lm
kit@nebula:~/prog/c$ tst
a[i][j] =
        11, 12, 13, 14, 15, 16
        21, 22, 23, 24, 25, 26
        31, 32, 33, 34, 35, 36
        41, 42, 43, 44, 45, 46
        51, 52, 53, 54, 55, 56
a[2][3] = 34
*(a[2]+3) = 34
*(*(a+2)+3) = 34
*(*a+2*n+3) = 34

```

It is true no two words are exactly the same. Synonyms are no exception. Here in Programme 1 both  $a[i][j]$  and  $*(*(a+i)+j)$  mean the same thing, the latter is more efficient than the former. This is because multiplication is computationally expensive, and  $a[i][j]$  where the dimension of  $a$  is  $m \times n$  actually means  $*(&a[0][0] + n \times i + j)$  whereas there is no multiplication in the case of  $*(*(a+i)+j)$ .

### Bibliography

Steven Holzner. *C Programming*. Brady, 1991